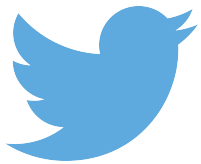
Sentiment Analysis in Twitter



Team Members

Rajarshi Sarkar (BE/1397/11)

Amit Kumar (BE/1513/11)

Project Mentor

Dr. Vijay Kumar Jha

IT8020 Project

Birla Institute of Technology, Mesra

Outline

What is Sentiment Analysis?

- Sentiment analysis refers to the use of natural language processing and text analysis to determine the attitude of a speaker or a writer. Example: Positive, Neutral or Negative. Happy, Sad or Angry, etc.

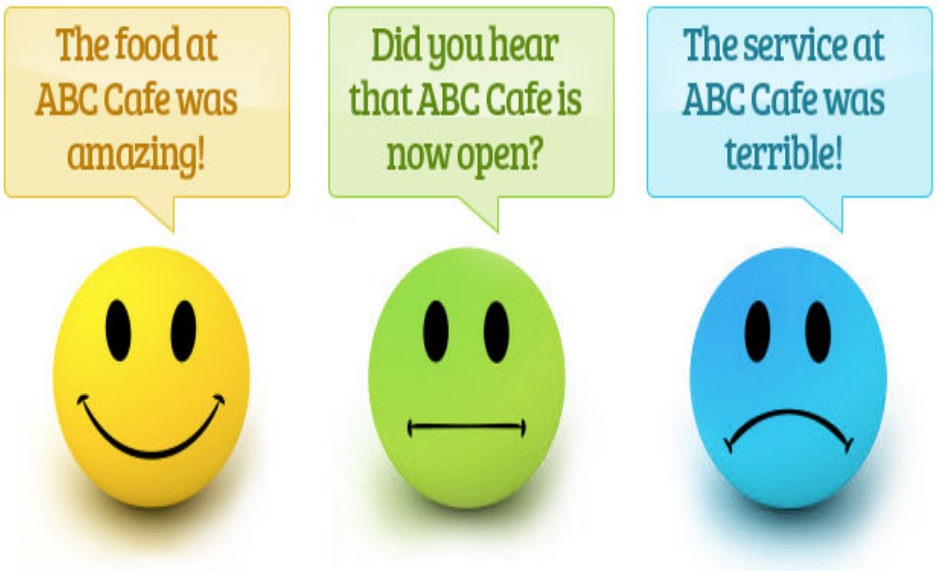
Tweet	**Sentiment**
I like cars	Positive
This is a car	Neutral
I hate cars	Negative

Why Sentiment Analysis in Twitter? [1] [2]

1. Tweets are short in length, i.e., they have less words.
2. Tweets can be fetched easily in JSON format.
3. Tweets of a specific topic can be fetched with help of a hashtag.

Objective

- The objective is to be able to automatically classify a tweet as a positive tweet or a neutral tweet or a negative tweet based on its sentiment.



Scope of the project

1. Consumers can use sentiment analysis to research products or services before making a purchase [3]. Eg: Kindle.
2. Marketers can use this to research public opinion of their company and products, or to analyze customer satisfaction. Eg: Election Polls [4].
3. Organizations can also use this to gather critical feedback about problems in newly released products. Eg: Brand Management (Nike, Adidas).

Tools used

Tools used are as follows:

1. **Python** [5]: It is a widely used general-purpose, high-level programming language.
2. **Natural Language Toolkit (NLTK)** [6]: It is a leading platform for building Python programs to work with human language data. It provides a suite of text processing libraries for classification, tokenization, etc.
3. **LIBSVM** [7]: A Library for Support Vector Machines.

Fetching Tweets

To fetch a tweet:

1. Create a twitter account if you do not already have one.
2. Go to https://dev.twitter.com/apps and log in with your twitter credentials and "Create New App".
3. Fill out the form and agree to the terms. This generates four keys: **api key, api secret key, access token key, and access secret key**.
4. Python code: Construct an oauth request, sign the request, and open a twitter request using the credentials above.
5. Python code: With the help of OpenerDirector instance of urllib module fetch the tweets in JSON format.
6. Python code: Tweets in JSON format having lang field as "en" are then processed further. The tweet is in "text" field.

Preprocessing Tweets

Preprocessing tweets includes:

1. Getting the tweet from the "text" field of a JSON document.
2. Convert the tweets to lower case.
3. Eliminate all of these URLs via regular expression matching or replace with generic word URL.
4. Eliminate '@username' via regex matching or replace it with generic word AT_USER.
5. Hashtags can give us some useful information, so it is useful to replace them with the exact same word without the hash. E.g. #nike replaced with 'nike'.
6. Remove punctuation at the start and ending of the tweets. E.g: 'the day is beautiful!' replaced with 'the day is beautiful'.
7. Remove additional white space.

Approaches used

1. Sentiment Dictionary Approach.
2. Naive Bayes Approach.
3. Support Vector Machines (SVM) Approach.

Sentiment Dictionary Approach: Overview

1. A file having 2482 words and their sentiment value (ranging from -5 to 5) is maintained. We call this file Sentiment Dictionary. [8]
2. A word and its sentiment in the Sentiment Dictionary is seperated by a tab character, i.e., '\t'.
3. A very positive word like 'happy' is given a sentiment value of 5.
4. A neutral word like 'aeroplane' is given a sentiment value close to 0.
5. A very negative word like 'hate' is given a sentiment value of -5.

Sentiment Dictionary Approach: Algorithm

The algorithm for Sentiment Dictionary Approach is as follows:

1. Get all the words present in the tweet.
2. Add the sentiment value of all words while referring to the Sentiment Dictionary. If the word is not present in the Sentiment Dictionary then add 0.
3. If the cumulative sentiment value is positive then the tweet is positive.
4. If the cumulative sentiment value is zero then the tweet is neutral.
5. If the cumulative sentiment value is negative then the tweet is negative.

Sentiment Dictionary Approach: Example

Words in tweet	I	like	cars
Sentiment value	0	2	0

Table: Cumulative sentiment value: 2. Tweet is classified as positive.

Words in tweet	My	name	is	Ram
Sentiment value	0	0	0	0

Table: Cumulative sentiment value: 0. Tweet is classified as neutral.

Words in tweet	I	hate	cars
Sentiment value	0	-3	0

Table: Cumulative sentiment value: -3. Tweet is classified as negative.

Sentiment Dictionary Approach: Problems

Problems in Sentiment Dictionary Approach are as follows:

1. If all the the words of the tweet are not present in the Sentiment Dictionary then the tweet will be judged as neutral which is not necessarily true always.
2. If a tweet has positive and negative words as per the Sentiment Dictionary, then they can very well add upto 0. Thereby, the tweet will be judged as neutral which is not necessarily true always.

Sentiment Dictionary Approach: Results

1000 testing tweets (367 positive, 418 neutral and 215 negative tweets) were tested against 2482 words of the Sentiment Dictionary. The results are tabulated below:

	Judged as		
	Positive	Neutral	Negative
Positive Testing Tweets (367)	**206**	152	9
Neutral Testing Tweets (418)	33	**358**	27
Negative Testing Tweets (215)	17	59	**139**

Table: Sentiment Dictionary Approach Analysis.

Overall Accuracy: 70.3 %

Naive Bayes Approach: Overview [9]

The data that has been successfully ported are as follows:

1. The Naive Bayes classifier is perhaps the simplest trained, probabilistic classifier model.
2. Also known as simple Bayes or independence Bayes.
3. Naive Bayes classifiers can be trained very efficiently in a supervised learning setting.
4. An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

Naive Bayes Approach: Algorithm [10]

1. Preprocess the testing tweets.
2. Remove all the stopwords (this, I, etc.) from the training and testing set.
3. Estimate the probability P(c) of each class c by dividing the number of words in tweets in c by the total number of words in the training data set.
4. Estimate the probability distribution P($w \mid c$) for all words w and classes c. This can be done by dividing the number of occurences of w in tweets in c by the total number of words in c.
5. To find the score of a tweet t for class c, calculate:

$$\text{score(t, c)} = \text{P(c)} * \prod_{i=1}^{n} \text{P}(w_i \mid c)$$

6. To predict the most likely class label, just pick the c with the highest score value.

Naive Bayes Approach: Example

There are 16905 words (5830, 5320, 5755 words in the positive, neutral and negative respectively) in the training data set.

Words in tweet	I	like	cars
Probability(positive)	-	8/5830	1/5830
Probability(neutral)	-	2/5320	3/5320
Probability(negative)	-	1/5755	1/5755

Positive score	5830/16905 * 8/5830 * 1/5830 = **0.000000081**
Neutral score	5320/16905 * 2/5320 * 3/5320 = 0.000000067
Negative score	5755/16905 * 1/5755 * 1/5755 = 0.000000010

Tweet classified as: Positive.

Naive Bayes Approach: Problems

Problems in Naive Bayes Approach are as follows:

1. It assumes each feature to be independent of all other features. This is the "naive" assumption seen in the multiplication of $P(w_i \mid c)$ in the definition of score. Thus, for example, if there is a feature 'best' and another 'world's best', then their probabilities would be multiplied as though independent, even though the two are overlapping.
2. The same issues arise for words that are highly correlated with other words (idioms, phrases, etc.).
3. Training time is high.
4. If there is no occurence of a word in a class and the word is present in the test tweet then after multiplication the score of the tweet will come as 0.

Naive Bayes Approach: Results

1000 testing tweets (367 positive, 418 neutral and 215 negative tweets) were tested against 1500 tweets [11] (500 positive, 500 neutral and 500 negative tweets). The results are tabulated below:

	Judged as		
	Positive	Neutral	Negative
Positive Testing Tweets (367)	**194**	89	84
Neutral Testing Tweets (418)	24	**347**	47
Negative Testing Tweets (215)	21	53	**141**

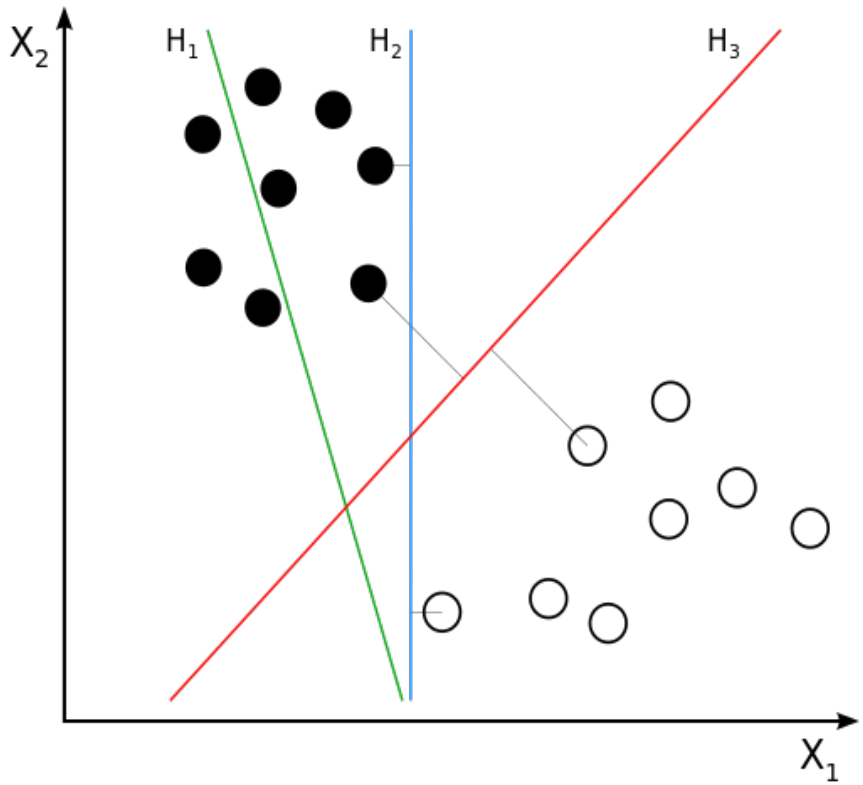
Table: Naive Bayes Approach Analysis.

Overall Accuracy: 68.2 %

Support Vector Machines Approach: Overview [12]

- Supervised learning model with associated learning algorithms that analyzes data and is used for classification.
- A data point is viewed as a p dimensional vector, and we want to know whether we can separate such points with a (p-1) dimensional hyperplane.
- The best hyperplane is the one that represents the largest separation, or margin, between the two classes.
- New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Support Vector Machines Approach: Overview [12]



H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximum margin.

Support Vector Machines Approach: Overview [12]

- Given some training data $\mathcal{D}$, a set of n points of the form:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^{n}$$

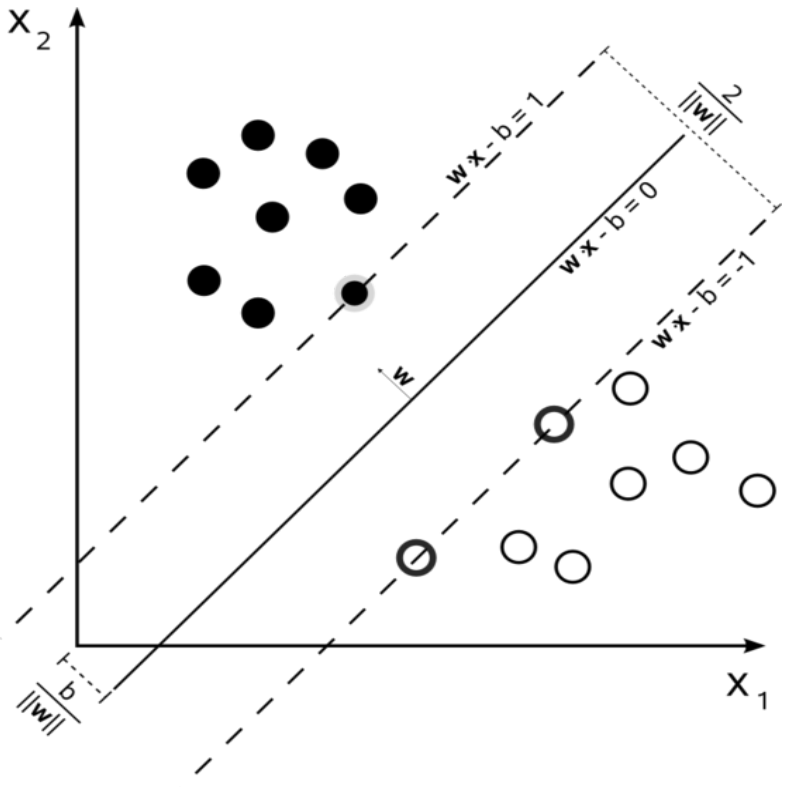
- Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying:

$$\mathbf{w} \cdot \mathbf{x} - b = 0.$$

  where $\cdot$ denotes the dot product and $\mathbf{w}$ the normal vector to the hyperplane.

- By using geometry, we find the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$, so we want to minimize $\|\mathbf{w}\|$.

Support Vector Machines Approach: Overview [12]



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

Support Vector Machines Approach: Overview [12]

- To minimize $\|\mathbf{w}\|$, $\|\mathbf{w}\|^2$ is minimized.
- This is a quadratic programming optimization problem:

$$\arg\min_{(\mathbf{w},b)} \|\mathbf{w}\|^2$$
$$\text{subject to } y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1$$
$$\text{(for any i} = 1, \ldots, \text{n)}.$$

- By introducing Lagrange multipliers $\boldsymbol{\alpha}$:

$$\arg\min_{(\mathbf{w},b)} \max_{(\boldsymbol{\alpha} \geq 0)} \{\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x_i} - b) - 1]\}$$

- This problem can now be solved by standard quadratic programming techniques leading to:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i}.$$

Support Vector Machines Approach: Algorithm

Algorithm implemented using LIBSVM is shown below:

1. Train the linear multiclass SVM classifier based on training tweet data set [11]. Each training tweet is mapped into the hyperplane with help of its feature list.
2. Preprocess the testing tweets and build a feature list for each tweet.
3. Map each testing tweet into the hyperplane with help of its feature list to know the class of the tweet.

Support Vector Machines Approach: Problems

Problems in Support Vector Machines Approach are as follows:

1. Parameters of a solved model were difficult to interpret.
2. Length of feature list is huge.

Support Vector Machines Approach: Results

1000 testing tweets (367 positive, 418 neutral and 215 negative tweets) were tested against 4550 tweets (2441 positive, 689 neutral and 1720 negative tweets). The results are tabulated below:

	Judged as		
	Positive	Neutral	Negative
Positive Testing Tweets (367)	**272**	53	42
Neutral Testing Tweets (418)	73	**332**	13
Negative Testing Tweets (215)	23	39	**153**

Table: Support Vector Machines Approach Analysis.

Overall Accuracy: 75.7 %

Future Work

Some of the future work to be done are as follows:

1. Using maximum entropy classifier and random forest [13].
2. Internationalisation [14]: Classify tweets of all languages.
3. Semantics [15]: Focusing on the relation between words, phrases, etc. Eg: 'Australia beats India' is positive for Australia and negative for India.
4. Bi-grams and Tri-grams [16]: Taking two or three words into consideration at the same time.

Conclusion

- Accuracy of classification depends on the training dataset.
- Accuracy of classification depends on the method used to classify.
- Using a novel feature vector of weighted unigrams Support Vector Machines can achieve competitive accuracy in classifying tweet sentiment.

Following is the summary of the accuracies of all the approaches used:

Approach used	**Accuracy**
Sentiment Dictionary	70.3 %
Naive Bayes	68.2 %
Support Vector Machines	75.7 %

References I

[1] S. Chinthala, R. Mande, S. Manne, and S. Vemuri, "Sentiment analysis on twitter streaming data," in *Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1* (S. C. Satapathy, A. Govardhan, K. S. Raju, and J. K. Mandal, eds.), vol. 337 of *Advances in Intelligent Systems and Computing*, pp. 161–168, Springer International Publishing, 2015.

[2] G. Paltoglou, "Sentiment analysis in social media," in *Online Collective Action* (N. Agarwal, M. Lim, and R. T. Wigand, eds.), Lecture Notes in Social Networks, pp. 3–17, Springer Vienna, 2014.

References II

[3] J. Dickerson, V. Kagan, and V. Subrahmanian, "Using sentiment to detect bots on twitter: Are humans more opinionated than bots?," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pp. 620–627, Aug 2014.

[4] K. Singhal, B. Agrawal, and N. Mittal, "Modeling indian general elections: Sentiment analysis of political twitter data," in *Information Systems Design and Intelligent Applications* (J. K. Mandal, S. C. Satapathy, M. Kumar Sanyal, P. P. Sarkar, and A. Mukhopadhyay, eds.), vol. 339 of *Advances in Intelligent Systems and Computing*, pp. 469–477, Springer India, 2015.

References III

[5] "Python Documentation." https://www.python.org/doc/.
[Online; Last accessed on 07-April-2015].

[6] "Natural Language Toolkit, NLTK 3.0."
http://www.nltk.org/.
[Online; Last accessed on 06-April-2015].

[7] "LIBSVM." http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[Online; Last accessed on 06-April-2015].

[8] "Sentiment Dictionary." http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010.
[Online; Last accessed on 06-April-2015].

References IV

[9] "Naive Bayes Classifier." http://en.wikipedia.org/wiki/Naive_Bayes_classifier. [Online; Last accessed on 06-April-2015].

[10] "Naive Bayes Classifier." http://sentiment.christopherpotts.net/classifiers.html#nb. [Online; Last accessed on 06-April-2015].

[11] "Training Data Set." http://goo.gl/DUkjbQ. [Online; Last accessed on 07-April-2015].

[12] "Support Vector Machines." http://en.wikipedia.org/wiki/Support_vector_machine. [Online; Last accessed on 07-April-2015].

References V

[13] N. F. da Silva, E. R. Hruschka, and E. R. H. Jr., "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, no. 0, pp. 170 – 179, 2014.

[14] M. Erdmann, K. Ikeda, H. Ishizaki, G. Hattori, and Y. Takishima, "Feature based sentiment analysis of tweets in multiple languages," in *Web Information Systems Engineering – WISE 2014* (B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, and Y. Zhang, eds.), vol. 8787 of *Lecture Notes in Computer Science*, pp. 109–124, Springer International Publishing, 2014.

References VI

[15] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of twitter," in *The Semantic Web – ISWC 2012* (P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, eds.), vol. 7649 of *Lecture Notes in Computer Science*, pp. 508–524, Springer Berlin Heidelberg, 2012.

[16] M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6266 – 6282, 2013.

Thank You!